Computer Architecture: Graduate Lab

Anyesha Ghosh & Prateek Sahu

RRIP Cache Replacement

- Developed to give a better scan resistance than LRU for mixed workloads.
- Replaces cache blocks based on RRIP (Re-reference interval) policy.
- Re-reference interval approximates the time to the next reference of a given block in the future.
- Uses an M-bit counter for each cache block, with larger values of M giving more granular recency tracking.

Insertion policy

- To insert a new block, find a victim block using the victim selection policy on the next slide.
- Replace the victim block by the new block.
- Set the RRIP bits (aka RRPV) of the new block to k. k can be any value between 0 (immediate re-reference) and pow(2,M)-1 (distant re-reference).
- The paper shows that an RRPV of pow(2,M)-2 gives the best results.

Victim selection policy

- To find a victim block, perform a scan for a block with RRPV=pow(2,M)-1. The scan can be performed from any starting point.
- If no such block is found, increase the RRPV for all blocks in the set by 1, and repeat the search.
- Repeat the above step till a block is found with RRPV=pow(2,M)-1.

RRPV Update on hit

- On a hit, the RRPV of the block is updated using one of two update strategies.
- Hit Policy: RRPV(*hit_block*) <- 0 (immediate r-reference)
- Frequency Policy: RRPV (*hit_block*) <- RRPV(*hit_block*)-1
- Frequency policy provides more accurate age tracking.

RRPV Update on invalidate

On Invalidate: RRPV(*invalidate_block*) <- pow(2,M)-1

RRIP vs LRU: Metrics

- Caches are evaluated on some common metrics:
- Hit Rate
- Hit access time given a constant capacity
- Hardware cost

- Here, we evaluate RRIP using the average hit rate (or alternatively the miss rate).

Methodology Used

- System simulated using the gem5 simulator.
- Simpoints used to collect representative intervals from the program execution.
- Data collected for five benckmarks: bzip2, hmmer, mcf, sphinx3, cactusADM.

Description of simpoints:

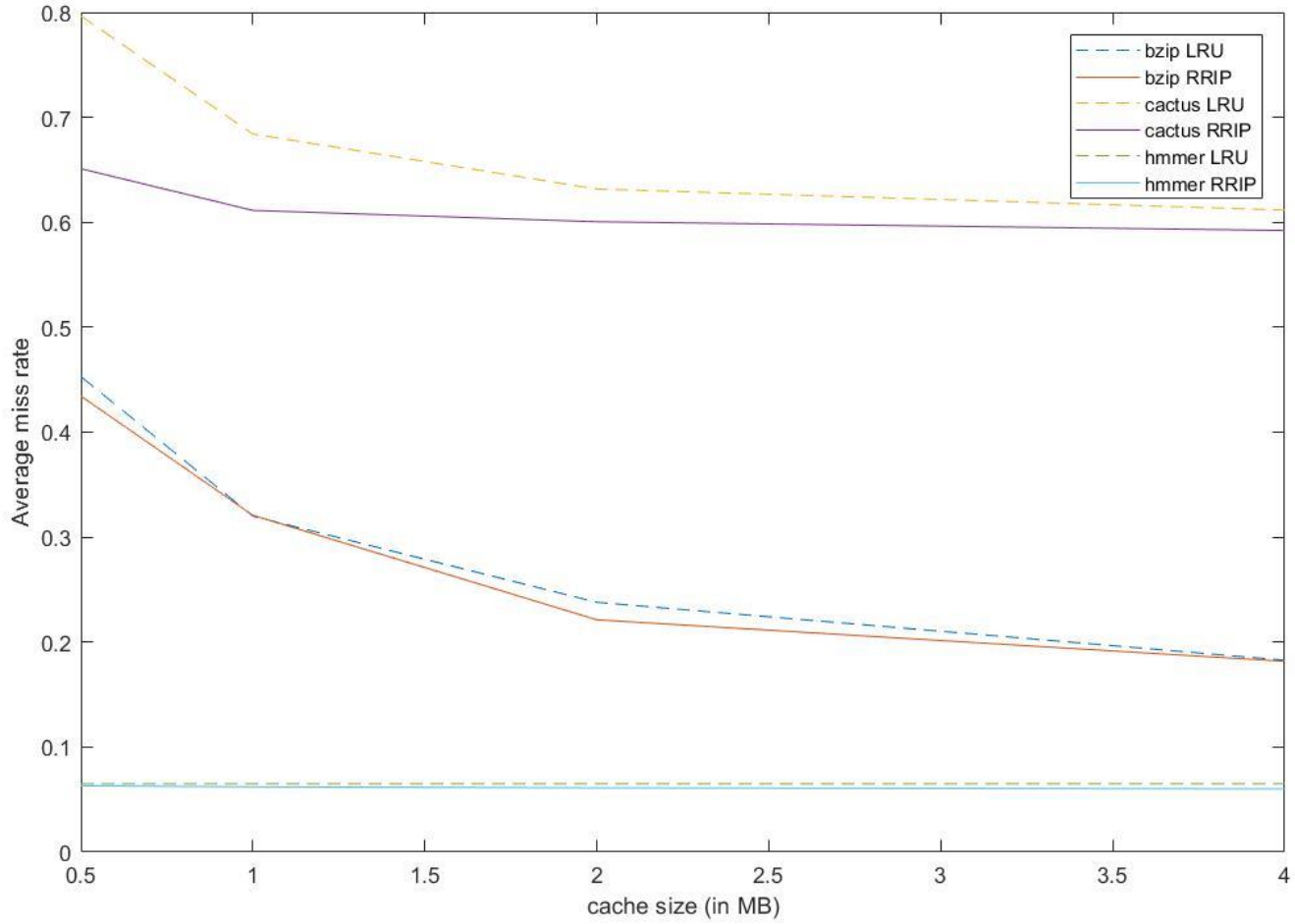
- Num. of intervals: 16
- Instructions per interval: 100 million
- Warmup cycles per interval: 1 million

System configuration

- Single CPU, 2 level cache hierarchy.
- Separate L1 Instruction & Data caches: (each) 32kB, 4 way set associative, LRU replacement policy.
- Large L2 cache: RRIP tested on this cache, so it has varying parameters.

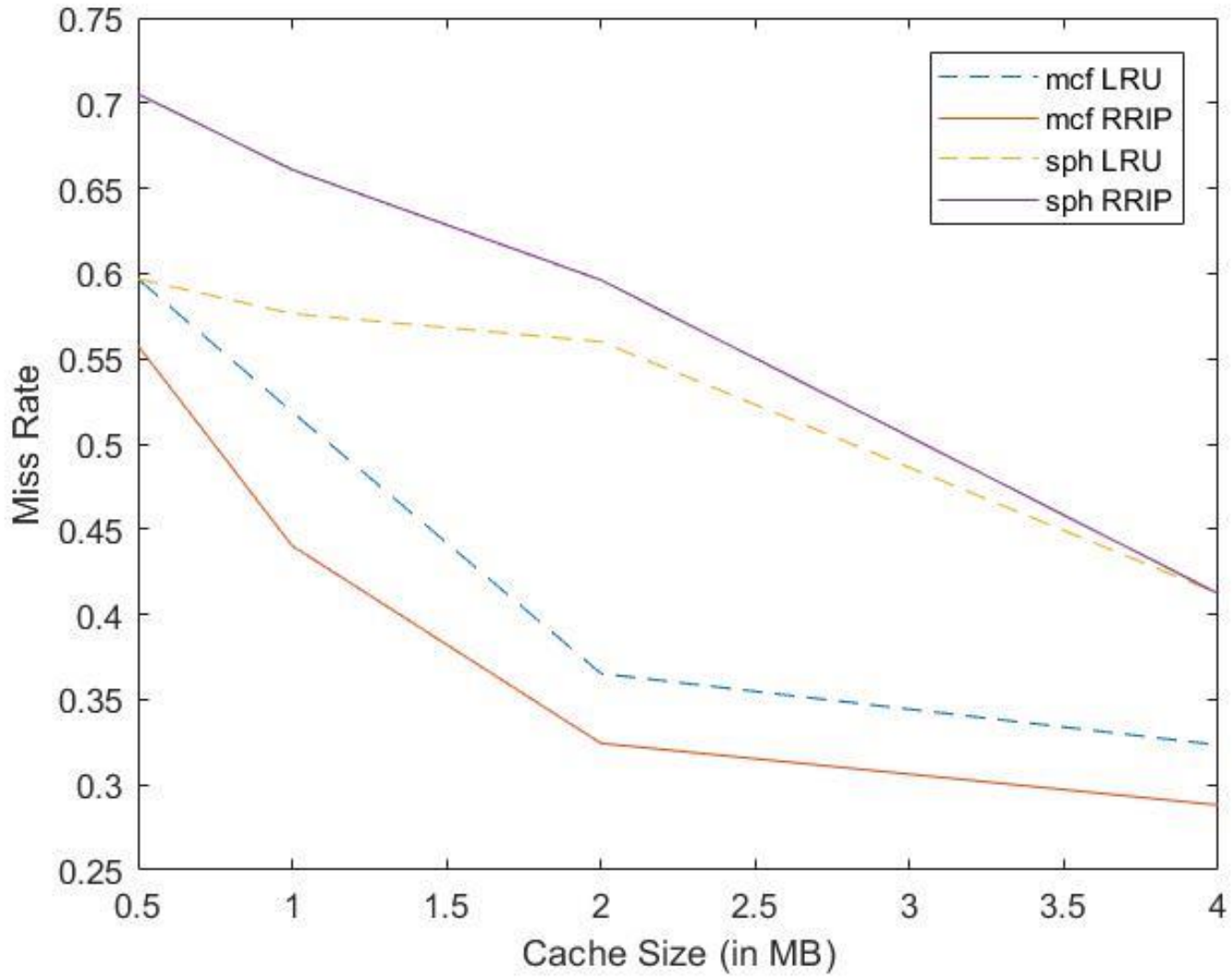
- Based on the recommendations of the paper, we decided to apply RRIP only on the LLC, as it doesn't give any benefit on lower level caches.

Plots: Varying cache size(1)



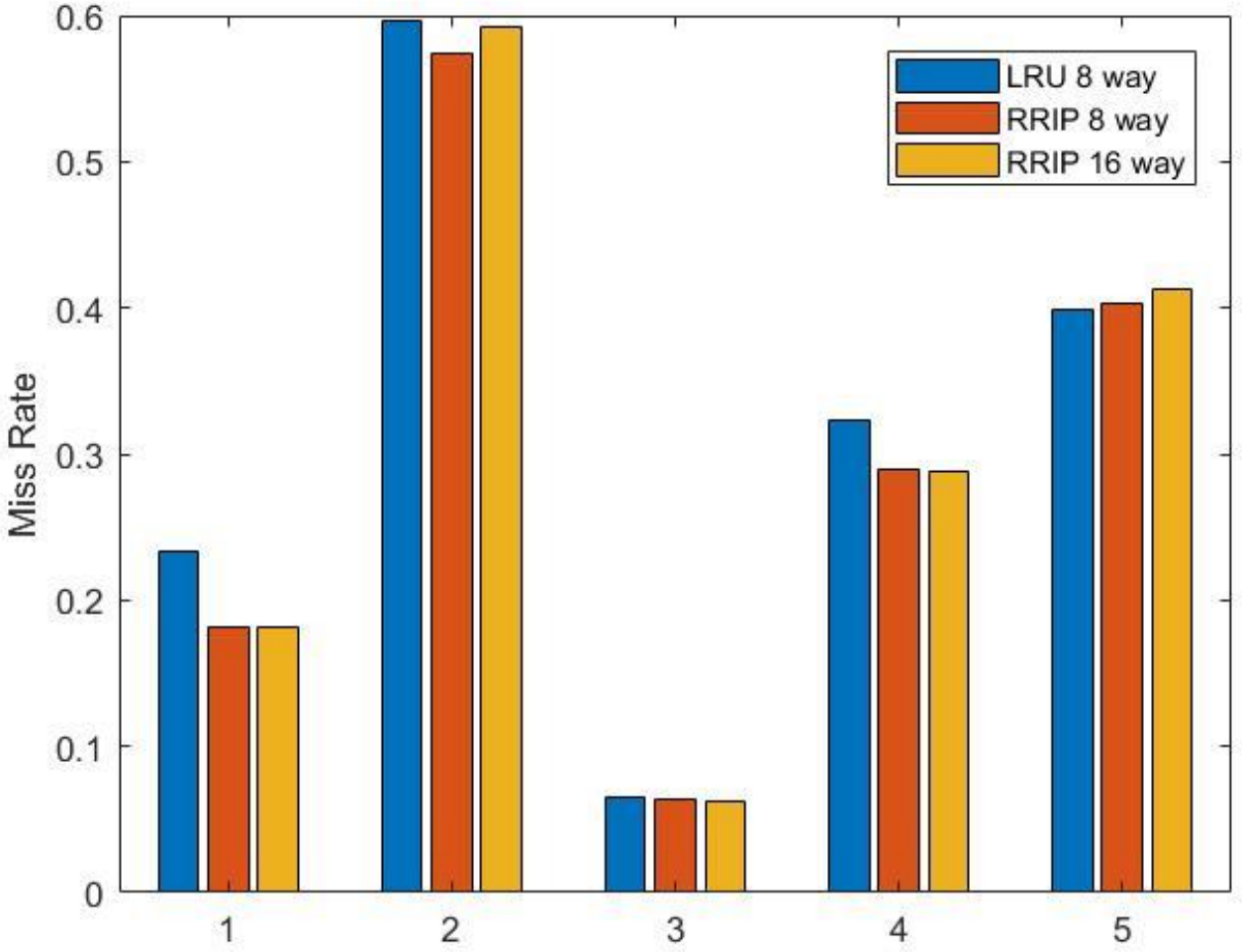
Cache configs: Hit policy for RRIP, 16 way set associative, 2b M value.

Plots: Varying Cache Size(2)



Cache configs: Hit policy for RRIP, 16 way set associative, 2b M value.

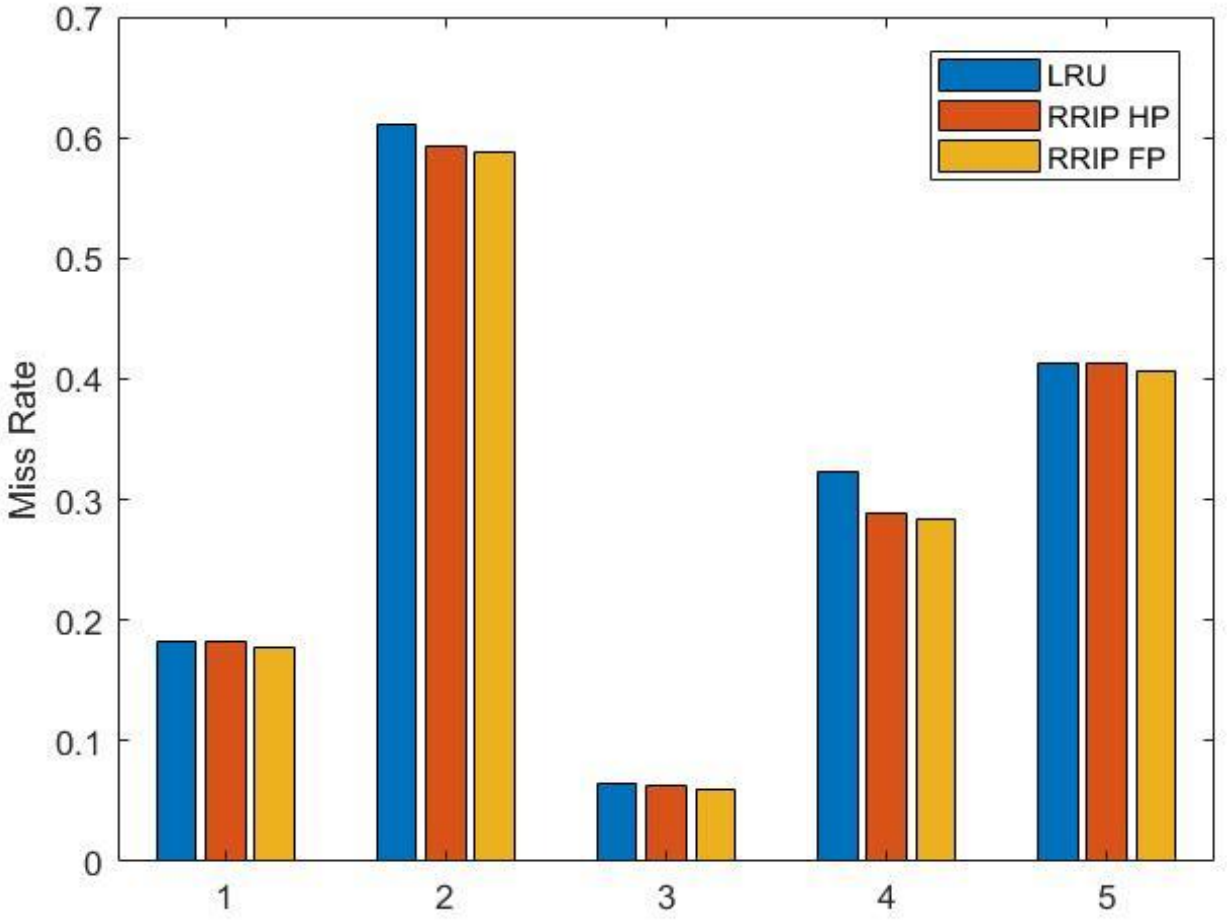
Plots: Varying Associativity



Cache configs: Hit policy for RRIP, 4MB cache size, 2b M value.

Benchmarks:
1: bzip2
2: cactusADM
3: hmmer
4: mcf
5: sphinx3

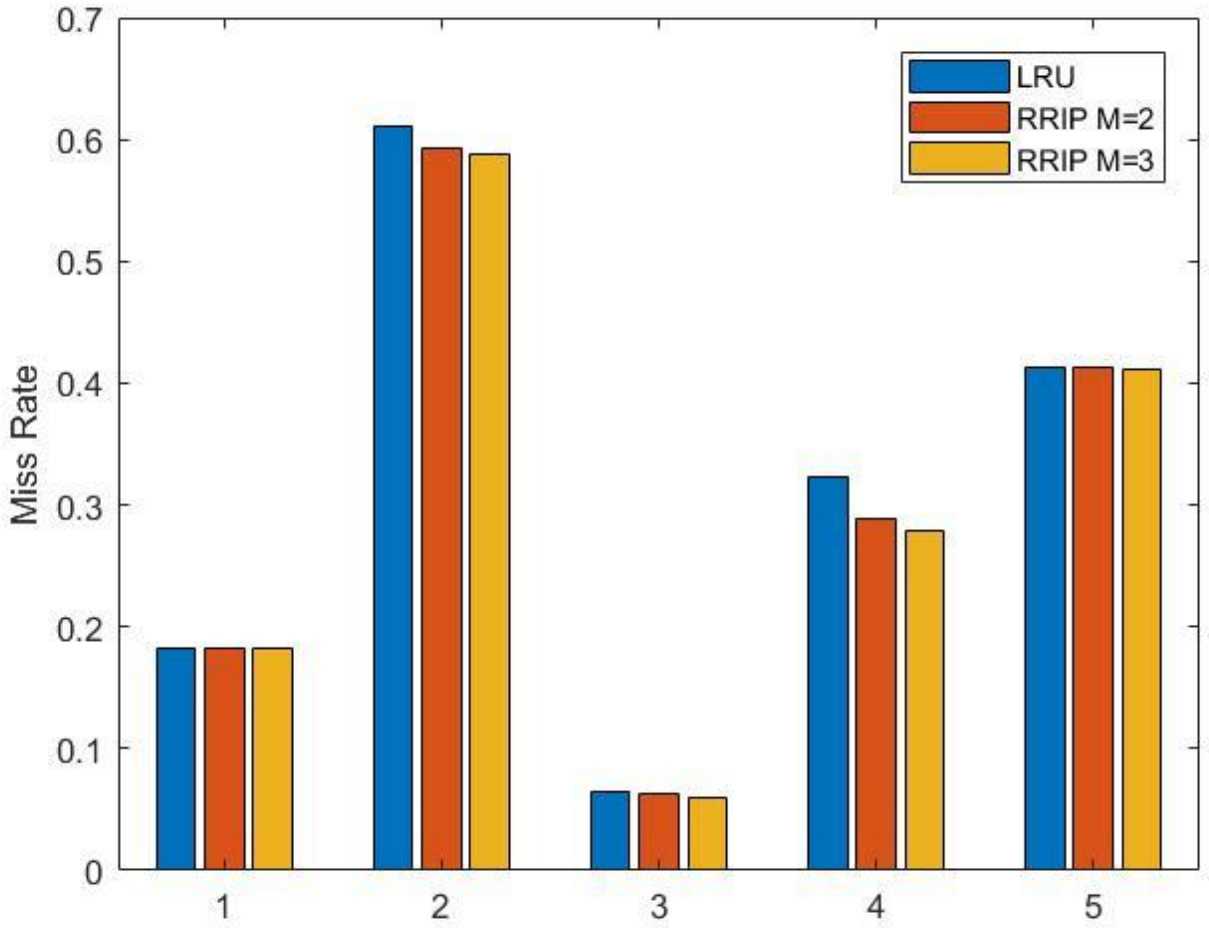
Plots: Varying Hit Policy



Cache configs: 4 KB cache size, 16 way set associative, 2b M value.

Benchmarks:
1: bzip2
2: cactusADM
3: hmmer
4: mcf
5: sphinx3

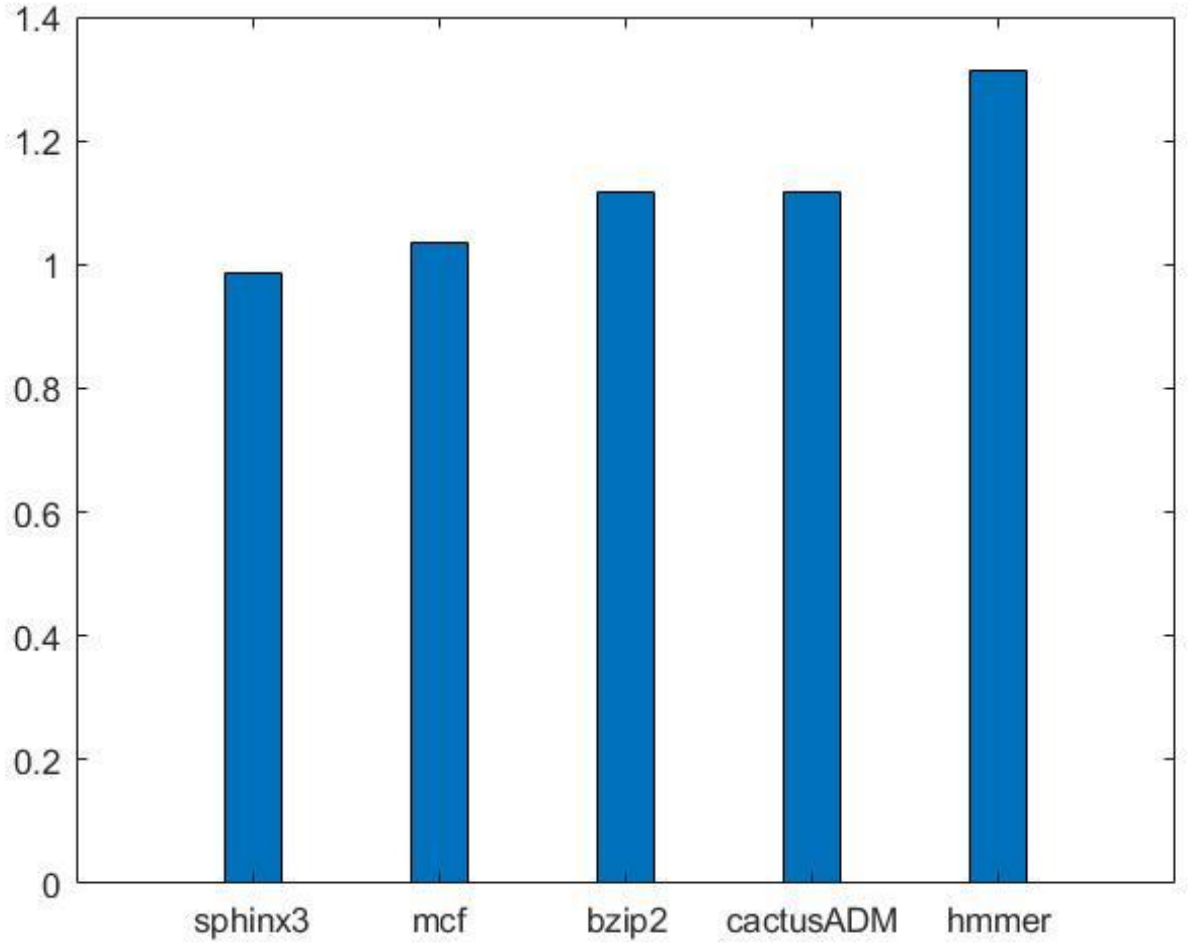
Plots: Effect of M



Cache configs: 4 KB cache size, 16 way set associative, Hit policy for RRIP.

Benchmarks:
1: bzip2
2: cactusADM
3: hmmer
4: mcf
5: sphinx3

S-Curve



Cache configs: 4 KB cache size, 16 way set associative, Hit policy for RRIP.

Y-axis shows:
miss_rate_LRU/miss_rate_RRIP

Observations

- RRIP gives a clear improvement over LRU for all the benchmarks, except for Sphinx. The anomalous result for Sphinx is probably due to the access patterns in the sampled intervals.
- The improvement provided by RRIP over LRU is greatest at a cache size of around 2MB across benchmarks.
- Varying associativity gives almost no measurable improvement. Any improvement by varying associativity would be solely due to the access pattern, and would not depend on the policy being chosen. This can be seen in the lack of a clear trend across benchmarks.

Observations (contd.)

- Changing the update policy to Frequency Policy (FP) gives a small reduction in the miss rate. This is expected due to the more effective age tracking it provides.
- Changing the width of M gives a small improvement in the miss rate. However, it also requires more status bits per cache block, and hence, more hardware.

Recommendations & Conclusion

- This study shows that RRIP is a good replacement policy for last level caches.
- We see that it gives good benefits for large LLCs at a minimal hardware overhead.
- From our observations, we conclude that RRIP is a viable & useful cache replacement policy.